

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

Кафедра автоматизации обработки информации

**УПРАВЛЕНИЕ ИЗМЕНЕНИЯМИ И РЕЛИЗАМИ ПРОГРАММНЫХ
СИСТЕМ**

Методические указания к практическим занятиям и организации
самостоятельной работы для студентов направления
«Программная инженерия» (уровень магистратура)

Томск, 2018

Бараксанов Дмитрий Николаевич

Управление изменениями и релизами программных систем:
Методические указания к практическим занятиям и организации
самостоятельной работы для студентов направления
«Программная инженерия» (уровень магистратура) / Д.Н. Бараксанов. – Томск,
2018. – 12 с.

© Томский государственный университет систем
управления и радиоэлектроники, 2018

© Бараксанов Д.Н., 2018

Содержание

1. Основные положения.....	4
2. Методические указания к проведению практических занятий.....	5
2.1. Практическое занятие «Управление версиями исходных кодов по методологии gitflow».....	5
2.2. Практическое занятие «Работа с docker контейнерами».....	6
2.3. Практическое занятие «Построение процесса непрерывной интеграции в Jenkins».....	6
2.4. Практическое занятие «Развертывание docker контейнеров в среде Kubernetes».....	7
2.5. Практическое занятие «Изучение систем управления задачами разработки программных систем и ведения проектной документации».....	8
2.6. Практическое занятие «Платформа как услуга: выбор решения».....	9
3. Самостоятельная работа.....	9
3.1. Самостоятельное изучение тем (вопросов) теоретической части курса...10	
3.2. Написание рефератов.....	10
4. Рекомендуемая литература.....	12

1. Основные положения

Целью изучения дисциплины «Планирование и организация вывода программного продукта на рынок» является формирование у студентов знаний о процессе планирования продвижения программного продукта, моделях поддержки принятия решений на разных стадиях планирования продвижения и практических навыков организации продвижения программного продукта на рынок с использованием современных средств коммуникационного воздействия на представителей целевой аудитории.

Цель лабораторных работ и самостоятельной работы – закрепить знания, полученные студентами на лекциях, получить практические навыки по планированию и организации продвижения программных продуктов (ПП) на рынок, научить студентов самостоятельно использовать комплекс моделей и алгоритмов поддержки принятия решений на разных стадиях организации процесса продвижения программных продуктов для решения прикладных практических задач.

При выполнении лабораторных занятий студентами усваиваются и закрепляются отдельные разделы курса по планированию и организации продвижения программных продуктов на рынок. Лабораторные работы выполняются на ПЭВМ, результаты работ оформляются в виде отчетов. Защита отчетов по лабораторным работам предусматривает как объяснение полученных результатов, так и ответы по разделам теории, на основе которой они получены.

Самостоятельная работа предполагает проработку лекционного материала, оформление отчетов по лабораторным работам, самостоятельное изучение отдельных тем теоретической части курса, а также выполнение индивидуальных заданий.

Преимущественно практические занятия проходят с использованием интерактивной обучающей платформы <https://www.katacoda.com/> [1].

2. Методические указания к проведению практических занятий

2.1. Практическое занятие «Управление версиями исходных кодов по методологии gitflow»

Цель работы: получить практические навыки управления исходным кодом программных продуктов по методологии gitflow.

Рекомендации по подготовке к занятию

При подготовке к занятию рекомендуется познакомиться с методологией gitflow используя открытые источники информации, рассмотреть основные команды системы контроля версий git.

Порядок выполнения задания:

1. Инициализировать новый git репозиторий, осуществить commit файлов.
2. Сделать изменения в файлах, просмотреть изменения с использованием git.
3. Осуществить push файлов в удаленный репозиторий.
4. Отменить последние изменения файлов до commit, а также отменить commit.
5. Создать отдельные features branches, научиться осуществлять merge с develop веткой. Сделать конфликтную ситуацию при слиянии веток, осуществить разрешение конфликтов.
6. Реализовать полный цикл управление ветками по методологии gitflow, включая тэгирование релиза и работу с hotfix ветками.

Форма проведения

Выполнять задание рекомендуется с использованием интерактивной обучающей платформы <https://www.katacoda.com/courses/git> [1]. В случае наличия установленного git-клиента на компьютере возможно выполнение задания с его использованием и публикацией репозитория во внешние сервисы

(например, github, gitlab и т. д.).

2.2. Практическое занятие «Работа с docker контейнерами»

Цель работы: получить практические навыки построения, настройки и запуска docker контейнеров.

Рекомендации по подготовке к занятию

При подготовке к занятию рекомендуется получить теоретические знания по микросервисной архитектуре программных систем и существующим технологиям виртуализации.

Порядок выполнения задания:

1. Запуск контейнера с помощью docker.
2. Запуск статичного HTML сайта в docker контейнере.
3. Построение docker образов.
4. Запуск node.js приложений в docker.
5. Запуск приложений с микросервисной архитектурой с использованием Docker Compose.

Форма проведения

Задание выполняется с использованием сценариев интерактивной обучающей платформы <https://www.katacoda.com/courses/docker> [1].

2.3. Практическое занятие «Построение процесса непрерывной интеграции в Jenkins»

Цель работы: получить практические навыки построения процесса непрерывной интеграции релизов программных продуктов используя продукт Jenkins.

Рекомендации по подготовке к занятию

При подготовке к занятию рекомендуется познакомиться с гибкими методологиями разработки программного обеспечения, включая требования к

обеспечению непрерывной интеграции и доставки релизов программных продуктов в среду промышленной эксплуатации.

Порядок выполнения задания:

1. Разработать pipeline сборки docker образа.

Форма проведения

Задание выполняется с использованием сценариев интерактивной обучающей платформы <https://www.katacoda.com/courses/jenkins> [1].

2.4. Практическое занятие «Развертывание docker контейнеров в среде Kubernetes»

Цель работы: получить практические навыки развертывания релизов программных систем с микросервисной архитектурой в среде исполнения docker контейнеров Kubernetes.

Рекомендации по подготовке к занятию

При подготовке к занятию рекомендуется познакомиться с возможностями сред исполнения docker контейнеров Kubernetes, OpenShift.

Порядок выполнения задания:

1. Развертывание контейнеров с помощью Kubectl.
2. Развертывание контейнеров с использованием YAML шаблонов.
3. Развертывание примера гостевой книги в Kubernetes.
4. Настройки маршрутизации на основе хоста и пути.
5. Настройка мониторинга «здоровья» развернутых релизов приложений.
6. Развертывание релиза программной системы в Kubernetes из исходных кодов.

Форма проведения

Задание выполняется с использованием сценариев интерактивной

обучающей платформы <https://www.katacoda.com/courses/kubernetes> [1].

2.5. Практическое занятие «Изучение систем управления задачами разработки программных систем и ведения проектной документации»

Цель работы: получить практические навыки использования автоматизированных систем учета и ведения задач разработки релизов программных систем, включая возможности ведения проектной документации.

Рекомендации по подготовке к занятию

При подготовке к занятию рекомендуется познакомиться с гибкими методологиями разработки программного обеспечения, включая организацию спринтов и требования к документации и периодичности выпуска релизов.

Порядок выполнения задания:

1. В ходе группового обсуждения формулируются основные требования и критерии оценки существующих решений поддержки процесса разработки программного обеспечения.

2. Учащиеся делятся по подгруппам и, используя открытые источники Интернет, подбирают по 3 схожие по функциональности системы.

3. Используя демо-доступ, отзывы пользователей, официальную документацию осуществляется анализ выбранных систем.

4. Результаты анализа оформляются в виде отчета, а также готовится презентация выбранных систем с рекомендациями по выбору одной из анализируемых.

5. Каждая подгруппа выступает с презентацией и представляет анализируемые системы.

6. На основе докладов и в ходе группового обсуждения формулируются обобщенные рекомендации по использованию систем.

Форма проведения

Основная часть занятий проводится в виде работы студентов по

подгруппам, групповые обсуждения проводятся в форме семинара.

2.6. Практическое занятие «Платформа как услуга: выбор решения»

Цель работы: получить практические навыки выбора и использования Platform as a Service (PaaS, платформа как услуга).

Рекомендации по подготовке к занятию

При подготовке к занятию рекомендуется ознакомиться с обзорами площадок, предоставляющими PaaS.

Порядок выполнения задания:

1. В ходе группового обсуждения формулируются основные требования и критерии оценки существующих PaaS площадок.

2. Учащиеся делятся по подгруппам и, используя открытые источники Интернет, подбирают по одной платформе.

3. Используя официальную документацию, демо-доступ, отзывы пользователей, готовится презентация платформы.

4. Изучаются сервисы платформы и осуществляется разворачивание тестового приложения в анализируемой платформе.

5. Каждая подгруппа выступает с презентацией и представляет анализируемые платформы.

6. На основе докладов и в ходе группового обсуждения формулируются обобщенные рекомендации по использованию PaaS.

Форма проведения

Основная часть занятий проводится в виде работы студентов по подгруппам, групповые обсуждения проводятся в форме семинара.

3. Самостоятельная работа

Целями самостоятельной работы являются систематизация, расширение и закрепление теоретических знаний в области различных аспектов организации релизного процесса и непрерывной интеграции в рамках

разработки программных продуктов.

Самостоятельная работа студента по дисциплине «Управление изменениями и релизами программных систем» включает следующие виды деятельности:

- 1) проработка лекционного материала;
- 2) подготовка к практическим занятиям;
- 3) самостоятельное изучение тем (вопросов) теоретической части курса;
- 4) написание рефератов.

В ходе самостоятельной работы студент, ориентируясь на изложенные рекомендации, планирует свое время и перечень необходимых работ в зависимости от индивидуальных психофизических особенностей. Формат самостоятельной работы студентов может отличаться в зависимости от формы обучения и объема аудиторной работы.

3.1. Самостоятельное изучение тем (вопросов) теоретической части курса

В ходе выполнения самостоятельной работы студенты должны изучить разделы «Подготовка инфраструктуры», «Тестирование пропускной способности», «Исследовательское тестирование» (стр. 52-110, 192-209, 210-218) [2].

3.2. Написание рефератов

Подготовка рефератов ориентирована на получение углубленных знаний по отдельным проблемам, затрагивающимся при изучении курса.

Примерные темы рефератов:

1. Обзор подходов к управлению релизами в различных методологиях разработки программных продуктов.
2. Обзор подходов к управлению изменениями в различных методологиях разработки программных продуктов.
3. Обзор продуктов непрерывной интеграции в разрезе

поддерживаемых языков и каркасов разработки программных продуктов.

4. Обзор продуктов мониторинга ИТ-инфраструктуры и исполняемых программных продуктов.

5. Обзор продуктов обеспечения логирования и мониторинга ошибок в процессе эксплуатации программных продуктов.

6. Обзор решений обеспечения интеграционного тестирования релизов программных продуктов.

7. Обзор метрик и решений оценки качества релизов программных продуктов.

8. Обзор решений поддержки управления релизами и изменениями программных продуктов.

4. Рекомендуемая литература

1. Interactive Learning and Training Platform for Software Engineers
[Электронный ресурс] - Режим доступа: <https://www.katacoda.com/>

2. Continuous delivery. Практика непрерывных апдейтов / В. Эберхард.
— СПб.: Питер, 2018. — 320 с.: ил. — (Серия «Для профессионалов»)
[Электронный ресурс] - Режим доступа: <https://ibooks.ru/reading.php?productid=356435>.

3. Философия DevOps. Искусство управления IT / Д. Дженнифер, Д.
Кэтрин. — СПб.: Питер, 2017. — 416 с.: ил. — (Серия «Бестселлеры O'Reilly»)
[Электронный ресурс] - Режим доступа: <https://ibooks.ru/reading.php?productid=355245>